AFRL-RH-WP-TR-2010-0105

Capability and Interface Assessment of Gaming Technologies for Future Multi-Unmanned Air Vehicle Systems

Marc Gacy
Alion Science and Technology

AUGUST 2011
Interim Report

Distribution A: Approved for public release; distribution unlimited.

See additional restrictions described on inside pages

AIR FORCE RESEARCH LABORATORY
711 HUMAN PERFORMANCE WING,
HUMAN EFFECTIVENESS DIRECTORATE,
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

AFRL-RH-WP-TR-2011-0105 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//signed//
Gloria Calhoun
Project Manager
Supervisory Control Interfaces Branch

//signed//
Gregory Barbato
Chief, Supervisory Control Interfaces Branch
Warfigher Interface Division

//signed//
Michael A. Stropki
Chief, Warfighter Interface Division
Human Effectiveness Directorate
711 Human Performance Wing

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
31-08-11	Interim	28 Jun 2010 – 4 May 2011

4. TITLE AND SUBTITLE
Capability and Interface Assessment of Gaming Technologies for Future Multi-Unmanned Air Vehicle Systems

5a. CONTRACT NUMBER
FA8650-08-D-6801

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62202F

6. AUTHOR(S)
Marc Gacy

5d. PROJECT NUMBER
7184

5e. TASK NUMBER
09

5f. WORK UNIT NUMBER
71840918

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Alion Science and Technology
4949 Pearl East Circle, Suite 200
Boulder, CO 80301

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Material Command
Air Force Research Laboratory
711th Human Performance Wing
Human Effectiveness Directorate
Warfighter Interface Division
System Control Interfaces Branch
Wright-Patterson Air Force Base, OH 45433-7511

10. SPONSOR/MONITOR'S ACRONYM(S)
711 HPW/RHCI

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RH-WP-TR-2011-0105

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution unlimited.

13. SUPPLEMENTARY NOTES
88 ABW/PA Cleared 10/21/2011; 88ABW-2011-5648.
This effort was conducted under Task Order 18: Human Supervision of Multiple Autonomous Vehicles in support of Ball Aerospace and Technologies Corp.

14. ABSTRACT
This report evaluates several gaming technologies and details their applicability to future control concepts for multiple unmanned air vehicle systems (UAS). Current gaming technologies evaluated include Real-Time Strategy (RTS) games, which require the simultaneous control of multiple entities; Massively Multiplayer Online Role Playing Games (MMORPG), which necessitate the management of multiple independent entities with sophisticated capabilities; and finally, arcade-style games, which demand the use of mitigation strategies to contend with variations in tempo. Accordingly, these technologies are then compared to current UAS control concepts, noting several key similarities and differences, to provide direction for future application.

15. SUBJECT TERMS
Unmanned Air Vehicle, Unmanned Air Vehicle Systems, Real-Time Strategy, Massively Multiplayer Online Role Playing Games, Situation Awareness, UAV, UAS, RTS, MMORPG, SA

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
U	U	U	SAR	34	Gloria Calhoun
					19b. TELEPONE NUMBER (***Include area code)***

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI-Std Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK.

TABLE OF CONTENTS

Section **Page**

LIST OF FIGURES

Figure **Page**

1.0 SUMMARY

This report details the study of gaming technologies with respect to their applicability to future control concepts for multiple unmanned air vehicle systems (UAS). A variety of concepts were considered, including screen layout, game-play tempo, teaming considerations, and automation behaviors among others. The results of the study were codified into a standalone database program which is searchable and editable, including having the capability of adding new records. During the course of the study, several key observations were made concerning gaming and UAS control systems. First and foremost, the popularity of Real-Time Strategy (RTS) games, such as *MechWarrior* and *Supreme Commander*, provide a consumer-tested platform for simultaneous control of multiple entities. Similarly, the popularity of Massively Multiplayer Online Role Playing Games (MMORPG), such as *World of Warcraft*, provide many concepts for coordination among independent entities with fairly sophisticated capabilities. Arcade –style games, such as *Tetris* and *Missile Command*, provide stripped down examples of increasing tempo considerations and mitigation strategies. Finally, we address several key differences between gaming and UAS control systems, both of which are important when transferring concepts from gaming to real world systems. One such notable difference is that games are designed to maximize the user's sense of engagement, rather than providing the most efficient means for accomplishing a given mission. Another notable difference is that games usually concentrate on one "style" of play (e.g., direct control or supervised control), whereas real systems require a fairly fluid transition among UAS control states. Additional differences, as well as the many prominent similarities, are highlighted with observations relating to individual games and suggestions for future research.

2.0 INTRODUCTION

Unmanned aerial vehicles (UAV) are becoming an increasingly critical aspect of military operations. With these successes comes an Air Force directed vision for more capable systems, including the ability of a *single operator* to control *multiple platforms,* and the desire for UAV accomplishment of more complex, dynamic missions in close collaboration with other manned and unmanned assets. Achieving this vision is predicated on a thorough understanding and proper design of the human-automation interface. Thus, there is a need to determine the most effective, over-arching architecture for implementing the automation-operator interface that establishes the rules for the transfer of control between the automation and the operator, as well as communication protocols, interaction styles, and cognitive strategies for reasoning with adjustable autonomy in operational contexts. An example of the control interface is shown in Figure 1, where an operator has a supervisory map with individual camera views, along with various status and control windows.

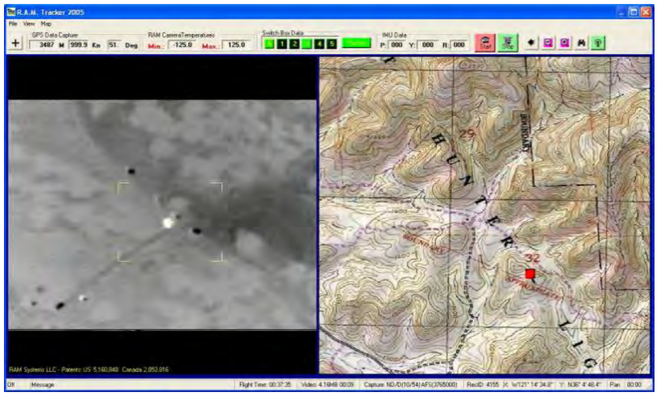

Figure 1. Radio Airborne Mapping (RAM) Tracker screen capture[1].

2.1 Hallmarks of Current UAV Control

Currently, UAV systems have a number of common control features from organic, platoon-level, hand-launched UAVs to long range reconnaissance platforms. First, such systems are at best 1:1 with one operator controlling one UAV. For more complex systems, the ratio can be several operators to one UAV. Also, UAV control and payload control often require separate operators (Figure 2). Additionally, control of a UAV is mentally demanding, such that operators often are limited to two to three hour shifts. Finally, due to the fact that UAV operation is still often direct control, trained pilots are often required for opeartion, especially with the larger systems. All of these will need to change for UAV systems control to advance to multi-UAV supervisory control.

Figure 2. OneSystem Universal Ground Control System (UGCS)[2].

2.2 Future UAV Control

There are numerous Air Force Research Laboratory (AFRL) programs that require automation allocation strategies for supervisory control of multiple UAVs. There is an immediate need to start considering the over-arching automation architecture that will drive the operator interface and interactions for these future systems. Such interaction will assume one operator controlling multiple UAVs, in either a supervisory or direct control capacity as needed, with either direct payload control or seamless handoff of control and/or results. The analysis of virtual system control capabilities and interface paradigms generated in this effort will greatly benefit these programs.

2.3 Design of Automation Architecture

A variety of approaches have been used in applying automation to complex systems. Traditional engineering mostly used the "*left-over principle*" for the allocation of functions, where the automated system was designed to do as much as was feasible and the rest was left for the operator. Human Factors Engineering introduced the *compensatory principle*, where human and machine capabilities are compared on salient criteria and function allocation is made such that the respective capabilities are used optimally. In the 1980's, with increasingly capable intelligent computing, the ideas of human-computer teamwork, and cognitive engineering, a *complementary principle* was considered in which function allocation serves to support and sustain human ability to perform efficiently. Here, focus shifts from human-automation interaction to human-automation cooperation; an "associate" architecture.

A key factor in how the operator interacts with the automation in this cooperation depends on what philosophy is used in building the automation architecture. The architecture defines the boundaries of the automation's functioning, determining the "freedom" with which it makes decisions, by considering constraints on the decision-making (limitations, rules, and regulations), decision-making abilities (authority, responsibility, and competency), and the capability to make different kinds of decisions (classes, functions, and levels). An example of such an approach is the operator serving as a supervisor and delegating the division of labor after determining what, when, and how the automated system will function.

2.4 Gaming Systems

In many ways, this automation architecture is similar to the approach taken by many current Real-Time Strategy (RTS) video games, where the player acts as a commander, delegating actions to the units, which are essentially autonomous agents in the computerized "world." First-Person Shooter (FPS) games have also become more sophisticated, integrating interactions with virtual team members to enhance the options available to the player (team leader). Understanding what capabilities the commander or team leader has at their disposal may indicate capabilities that are missing or need to be augmented in real UAV systems.

Video games have already been identified as a valuable source for interface design. The intelligent algorithms driving the action in many games might also be useful, as they are designed to work with the operator following a mixed-initiative philosophy. Examples include how one move helps set the stage for another move, and the various ways automation is used to help the player allocate resources, keep track of past movements, know what to do, and

coordinate actions to engage a dynamic target. It would also be useful, if possible, to learn how the gaming industry conducts its own analysis of whether the intelligent behaviors and actions programmed for its game are operating appropriately. There may be unique computation models, goal graphs, etc. that provide helpful visualization into the automation's functionality.

3.0 OBJECTIVE AND PROCEDURES

3.1 Objective

The objective of this effort is to evaluate past and current gaming approaches to determine their applicability to future multi-UAV control. In particular, we will address how these systems complement and augment advances in traditional autonomy.

Gaming systems are developed over a period of many years often at a cost of millions of dollars. The target audience usually has a great deal of experience with such games and a wide variety from which to choose. This pressure results in a great deal of effort to maximize not only the game-play, but also the user experience. Thus, it makes economical sense to evaluate successful systems and assess their applicability to real UAV systems. The results of this present effort will provide a rationale of why gaming systems do or do not show promise for application to UAV autonomy, and will identify basic research issues that need to be addressed before initiating applied research to develop concepts tailored towards multi-UAV supervisory control.

Our assessment will address a variety of current issues in associate-based UAV control, including, but not limited to: adapting the amount of information presented as a function of necessary involvement and workload; delegating or adapting control actions via a predetermined philosophy or set of guiding principles; and assessing operator's trust in decisions made by the virtual agents, even if they diverge from similar decisions made by the operators.

3.2 Procedures

The first stage of this project was to develop a database library of videogames. The second stage of the project was to select games based on their database entries and explore them further, through actual game-play, screenshot analysis, control paradigms, and reviewer comments.

4.0 DATABASE DEVELOPMENT

The purpose of the database is two-fold. The first is to create a fairly comprehensive list of game types to evaluate from a variety of genres. From there, a series of categories were developed that relate to various autonomous control concepts. Detailed descriptions of each of the categories and their entries are provided in the Appendix. The database was then populated with games and their relevance to selected topics. The structure of the database was purposefully kept "flat" (i.e., no nesting of categories) to allow for simple searching and retrieval. An example of the database is provided for the game *Raven Squad* (see Figure 3).

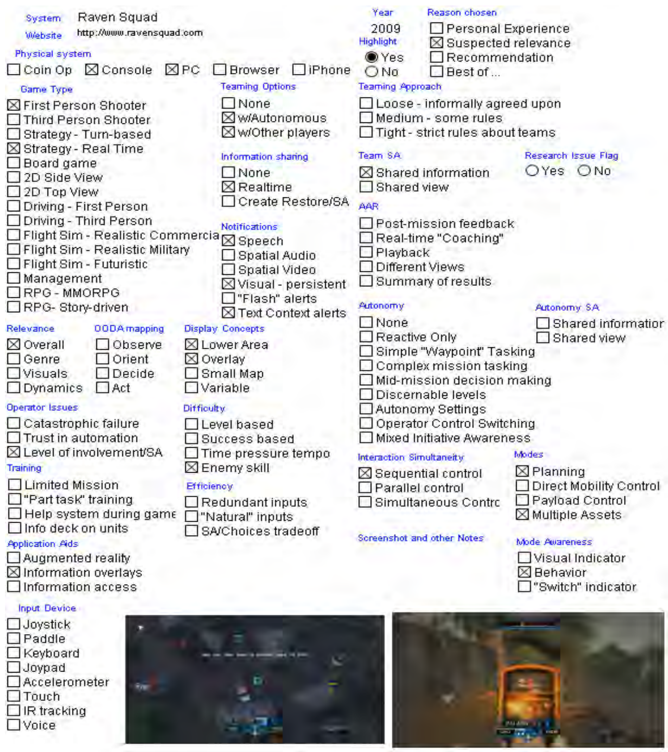

Figure 3. Games database screen capture.

Furthermore, the initial concept was expanded beyond just FPS and RTS games to encompass all genres of video games, since several of them have concepts that are relevant and sometimes in a "more pure" form than the more complex games. Examples of other types of games include: flight simulators (e.g., *UAV Predator)*, static defense games (e.g., *Missile Command*), resource management games (e.g., *Sim City 2000*), board game simulations (e.g., *VASSAL*), and abstract games (e.g., *Tetris*).

The second purpose of the database is to help identify games that address research concepts, and to provide a means for which to store screenshots and other relevant information. This design will allow the database to continue to have utility beyond the initial work by providing a method for simple database entry. Future work may then expand on the database with more examples, providing enough data to perform statistical analysis of games by correlating which types of games stress which features.

5.0 GAME EXPLORATION

During the course of the research, we noticed that a majority of the games fell into one of several broad categories, and only a few of these provided more than a cursory insight into multi-system control. As expected, RTS games provided many examples of both interface concepts and actual control concepts. FPS games, especially with multiplayer interaction options, also provided some valuable insight both into what is and what is not considered successful. Arcade-style games, while on the surface seem unrelated, often provided "pure" examples of various concepts, uncluttered by game complexity and graphics. Finally, turn-based games provided the

greatest level of time luxury, and thus most accurately represent the planning stages of UAV operation.

The following sections present several potential areas for research with respect to multi-UAV control. First, a control concept will be discussed, then a game that demonstrates that concept. Finally, a potential research area and how it relates to multi-UAV control will be discussed.

5.1 Operator Overload and Catastrophic Failure – *Mission Command* and *Tetris*

In single player, level-based video games (not story-driven) there is often a point at which the game becomes too difficult and the players will subsequently lose the game. For many games this is a gradual decline, where the loss of lives often occurs gradually over the course of the game, with the final loss indicating the end. There are several games, however, where the losses can come more quickly than suggested by the incremental increase in difficulty. These games represent a catastrophic failure on the part of the game player, not necessarily an actual "catastrophic" failure in the mathematical sense (e.g. functional discontinuities), but rather the conceptual definition of a sudden shift in behavior arising from a small change in circumstance. *Missile Command* and *Tetris* are good examples of this behavior and represent two simple ways in which such behavior can arise.

In *Missile Command*, the operator is using three independent missile silos to fire at and destroy incoming missiles (see Figure 4). The operator selects which silo is best to fire from based on the location of the incoming missiles. The object of the game is to keep the missiles from destroying both the missile silos and the cities on the ground during an attack wave. When players achieve a certain level of proficiency, they can often play for many levels without losing a "life," only to lose all or nearly all of the cities on the next level. Usually this occurs because the player lets a silo get destroyed, or they lose their "cadence" with placement of missile hit sites, which results in a number of enemy missiles getting through.

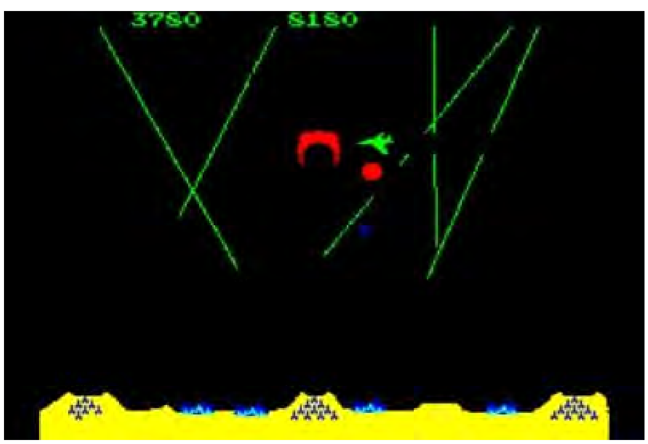

Figure 4. Missile Command screen capture[3].

Tetris has a similar effect; if the operator is unable to correctly place a game piece or makes a mistake in placement, subsequent placement of game pieces is difficult because of the reduced space available and an inability to remove the blockage (Figure 5).

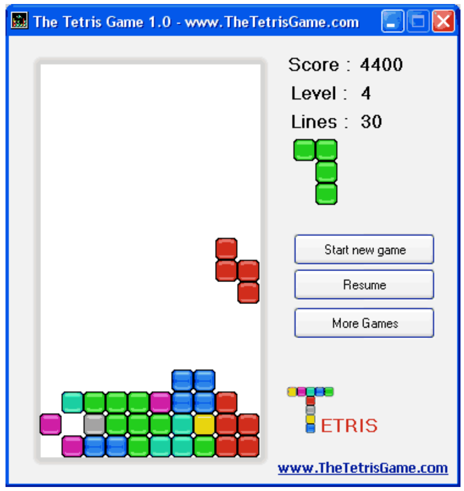

Figure 5. Tetris screen capture[4].

Both of these games are extremely popular, having versions in almost every electronic media category. Much of this can be attributed to the sense of accomplishment for keeping the cadence and staying ahead of the failure. Both games highlight this certain kind of effect in error propagation, where making a mistake makes subsequent mistakes more likely. This kind of behavior can lead to catastrophic failure whereas simple probabilistic errors rarely achieve such a collapse in performance.

A potential research area is whether or not the UAV system is prone to increasing error probabilities. If it is, can mitigation strategies, much like *Tetris*' presentation of the next piece, alleviate some or all of that behavior? To continue the *Tetris* example, if the system detects that an error occurred, for the next few drops it may suggest placement options or show the next two pieces to come, until a certain number of correct placements occur.

Note that this problem is fairly independent of the complexity of the system, and may even be accentuated by the repetition of simple tasks. Thus, this research area may be relevant to issues of control with a very large number of fairly autonomous UAVs where the operator is issuing simple commands, but very frequently.

5.2 Mode Awareness and Measuring Plan Progress of Units – *Starcraft*

The game genre that best represents the high-level control of multiple unmanned systems is the RTS genre, of which *Starcraft* is an early and successful example. The ability to manage a large number of units is facilitated by its interface with a summary of unit types and states with more detailed information upon selection. The units are also given rudimentary, single point commands, such as patrol (move back and forth between two points), move (from current position to specified position), attack, build, or repair.

The system provides auditory (verbal) feedback that the unit has been properly selected (mode selection) and that it has received the input command, but it does not do a good job of indicating that the units are following the command or that they have finished the task. Two common problematic situations are that the unit immediately encounters an obstacle and just stops or it decides to take a long circuitous path around obstacles. While this kind of behavior may not be particularly common with air vehicles, it is quite prominent with unmanned ground

systems. Such behavior might be present with UAV systems moving through cluttered or restricted space with the potential for unintentional rerouting around such spaces. The units also do not give any indication that they are finished with a task. Usually that determination is left up to the player by observing movement (or lack of it) on the small mini-map in the lower left corner (see Figure 6).

Figure 6. Starcraft screen capture and manual portion, showing minimap in the lower left corner, selected units or individual unit information in the center and command icons on the right[5,6].

A potential research area is how to best indicate that units have finished their task, or more importantly, somehow deviated from their task. Auditory indicators are a first suggestion, but perhaps they were not used due to the potential for too much auditory clutter. Auditory alerts are used to great effect in real-life combat scenarios to indicate when units are under attack; thus, plan deviations (another class of "unexpected behavior") might also be well suited to auditory alerting. In order to reduce auditory clutter, another mode, such as a visual indicator, might be better suited for the initial selection and tasking of the unit. There is a strong argument for using the visual channel to provide feedback considering the player already has visual contact with the unit when tasking.

It is also important to point out which unit is selected and that the selected unit is given the correct task. This is a subtler form of mode awareness, which can be disastrous when the wrong unit is sent on the wrong mission. Many times, as in *Starcraft*, only the unit type is referred to in the feedback; any identifier, special designation, or grouping is not used, furthering the potential for confusion. When controlling multiple UAVs, this is particularly likely to be a problem as the operator will be controlling a number of UAV systems with mostly identical capabilities. Subtle indicators, such as vehicle type or available/unavailable commands, do not

exist, leaving the operator to rely on name and memory of the UAV's position and mission alone for distinction.

It should be noted that a likely explanation for *not* including this behavior in games is that it has the potential to remove the operator as an active participant, such that they are simply responding to the various cues they receive as opposed to actually controlling the situation. This lack of engagement can often be translated to a loss of Situation Awareness (SA), which will be discussed in more detail below with the game *Raven Squad*.

5.3 Interface Concepts – *Mech Commander*

While some basic interface concepts are touched on briefly above, the game *Mech Commander* has a similar approach, but with more user control over the presentation. In *Mech Commander*, the player has the option of selecting Map, Data, Briefing, and Salvage, where Data is information about the selected Mech, Briefing is a summary of the mission, and Salvage is a record of picked up goods (inventory). The player may select any of these to be in the information window (shown in the upper left in Figure 7). The player may also choose to minimize the information window altogether, so more of the screen area is devoted to the main "battle" view. This was primarily done in response to complaints about static interfaces, which have the advantage of continuously providing information, but the disadvantage of taking up room that some feel is better served by more space in the main window. The definite advantage of the *Mech Commander* interface is that there is a large area for individual unit information, which can take up the entire information area, whereas in the static display, it is confined to a smaller area.

Figure 7. Mech Commander screen capture with information area in the upper left corner showing the map display[7].

This attention to unit status is also evident in the "traveling status bar" that accompanies each unit (rectangular bar above the unit, Figure 7). Similarly, the selection aspect of the game makes use of text overlays to provide specific information about the selected object, in addition to the usual color change selection among neutral (or uncommitted), friendly, or enemy targets.

These enhancements, which are present in many subsequent games, are examples of augmented reality, and are similar to what is viewed through a Heads-Up Display (HUD). While in a video game it is easy to keep the overlay information correctly registered with the player's view, in a real situation, especially with many assets and visual entities potentially cluttering the display, small errors in the location/registration of the overlay information may confound or confuse the operator instead of providing immediate status information.

"Attaching" information directly to the units, through a form of augmented reality, is certainly an attractive concept based on the evolution of RTS games. As described above, however, there is the potential for confusion with multiple real assets, particularly on a live map. A potential research area could address the minimum level of "persistent" or always available information to the operator, regardless if the UAV is in the current map view of the operator. Some questions that could be addressed are whether the alerting and representation behavior for those UAVs "off the map" should somehow be different for those UAVs that are "on the map," have status information immediately visible, and are readily selectable. Such research could address attempts to minimize the "out of sight, out of mind" effect for non-visible UAVs, and how to reduce the time to acknowledge and address a situation with "off the map" UAVs.

5.4 Teaming – *World of Warcraft*

World of Warcraft is the most popular Massively Multiplayer On-line Role Playing Game (MMORPG) and as such has incorporated many display control concepts from other games over the years. One area that is a key consideration for MMORPGs is teaming (with other human players). *World of Warcraft* has specific rules for teaming at two different levels: parties, which are small groups that comprise only a few members; and guilds, which can comprise many members. Though there is often overlap between the two, they serve two different functions. A party can be viewed as essentially an ad-hoc, task-based team structure where one leader (who creates the party) invites others to join a party. Parties are usually created for a single adventure (although parties can easily be used for multiple adventures) and the game structure is very specific about who the leader is, how communication is handled, and how "loot" is distributed. A guild is a larger, more permanent organization, with a stratified hierarchy and more stringent rules for both formation and maintenance. A raid is a special subset of a party, which is a large collection of players (officially up to 40, but sometimes upwards of 300 or more) working toward a common goal (see Figure 8). The complexities of keeping such a large number of individual players focused and efficient is interesting to study as the actual interaction rules for the structures are fairly loose, being primarily in the form of communication filters (party only or guild only) and some additional status indicators (see the right side of the screen in Figure 8).

Figure 8. World of Warcraft screen capture with an information bar and myriad overlays[8].

Furthermore, *World of Warcraft* has one of the richest display environments, with many interface concepts being present, either through the base interface or through user-customized add-ons. First, it has the bottom status bar, with a centralized mini-map. Several team features are provided as overlays, with raid members' status in green and a target list in blue (both on the right in Figure 8). Additionally, color coding is heavily used as both a primary (squares) and secondary (names list) indicator of information. Character names/groups are included as transparent words that move with the character, providing an "augmented reality" overlay to the view.

World of Warcraft and many other MMORPGs have several areas with the potential for valid research. First, the wide variety of interface concepts and the ability to create, add, and remove the concepts as desired provides the level of control over the interface needed for any comparative study. A good example of this type of add-on is called "Decursive[9]." This add-on provides small translucent squares on the left side of the screen (see Figure 9). These squares are color coded to provide relevant status information and are organized in descending order of importance: self, group, and raid group. These squares represent a form of augmented reality, in that they automatically assess the relevant status information of all the team members, and they also represent a limited form of autonomy, by automatically preparing a response to adverse status conditions. This allows the player to take on the role (at least for this particular function) of a supervisor authorizing the indicated actions. This approach could be pushed slightly in either direction to a more or less autonomous mode, by only indicating the status without the suggestion (less autonomy), and by automatically performing the requested action by default after a predetermined period of time concludes (more autonomy). Furthermore, intelligence could be built into the ordering process. Instead of just defaulting to hierarchical importance, the physical proximity to the character and/or the severity of the condition could also play a role. For control of multiple UAVs, these small differences in the level of autonomy could change from well coordinated missions to either unacceptably slow missions or unacceptable losses of SA. The level of autonomy change isn't great, but the effect compounded over many choices with many units can highlight the less optimal solutions. Using *World of Warcraft* as a research tool

may be different as there is a current rule and built-in limitation against purely automated behaviors, which encourages the maintenance of the human element in the game.

Figure 9. "Decursive" overlay screen capture showing colors and transparent overlay[10].

5.5 Situation Awareness – *Supreme Commander*

The geographic scale of multi-UAV control is well beyond that of most games. Due to the larger area covered by the UAVs, it may be expected that maintaining SA through graphic cues would be more difficult for UAV control than in video games. In many games, the "action" takes place in a well-defined area that often is indicated by a small overview map (described above). While in most cases this is true, there are a few games that take advantage of a much larger area. *Supreme Commander* has such a larger area of operations than most games, claiming to be theater-based, as opposed to battle-based. The player interacts with the map in a very fluid manner, being able to zoom in at any level from single-vehicle, third person close-ups, all the way out to the entire theater space (see Figure 10).

Figure 10. Supreme Commander screen capture zoomed out to theater level[11].

The intuitive and easy to use controls of the continuous zoom feature go a long way to maintaining SA by allowing the player to visually map the changes in zoom level. Furthermore, the player may choose to have anywhere from one to three maps, all independently zoomable, including a split screen and a small overview map (see Figure 11). Each is independently controllable and can provide both top-down or angled views (much like Google Earth). What is missing from the split screen maps is a way to easily relate them to each other, especially if there is no area of overlap. Ideally, one could argue that the maps should have a common overview map linking them for proper SA and registration.

Figure 11. Supreme Commander screen capture with split screen[12].

Also, *Supreme Commander* has another feature which allows the player to create waypoint paths for the selected units and to display those paths later through a toggling system (see Figure 12). This provides a much higher level of planning and execution SA than other systems while keeping the clutter to a minimum by only presenting the information when the player actively requests it.

Figure 12. Supreme Commander screen capture showing waypoints[13].

A potential research area, which builds off of this approach, is to determine how much information is necessary to convey the planned path when the viewer is not the person who created the path. Additional information may be required for the new operator to have appropriate SA, such as purpose, time started, and all units following the path. For multi-UAV control this is important as mission monitors may or may not be the same operators who develop the missions and initially pass them to the UAVs. The mission monitor should be able to query details of an entire plan, with explanations as needed for certain programmed behaviors.

5.6 Real-Time Strategy and First-Person Shooter Combination – *Raven Squad*

One area that indicates further research is needed was brought to our attention, not by the presence of a certain game, but rather by its relative absence. The combination of a RTS game with a (FPS) game would provide one of the most thorough overviews of unmanned system capabilities. However, this combination is rarely attempted, and when it is, the result is often a poorly received game.

Raven Squad, by many accounts, seems to be one of the first video games to really combine both RTS and FPS aspects into the main gameplay[14,15] (see Figures 13 & 14). Several other games, *Iron Grip: Warlord*, *Savage 2*, *Battlezone 2*, and *Natural Selection* also combine these two elements, although usually either one of the two aspects heavily dominates, or the game-play is very sequential. *Raven Squad*, however, allows a player to shift back and forth between the two aspects rather seamlessly, but the FPS aspect seems more engaging and the player often feels more in control. This factor, along with reviews that state the somewhat confusing nature of the game, provide two key areas for all, since for each round, the player elects to attack or move away, and the result is decided randomly based on the statistics of the asset.

Figure 13. Raven Squad screen capture in Real-Time Strategy Mode[16].

Figure 14. Raven Squad screen capture in First-Person Shooter Mode[17].

As suggested, the loss of SA or "confusion" that results from switching between modes is a real problem both in this game and in the real world. In *Raven Squad*, even though it is fairly simple to switch, there are a limited number of links to facilitate the transition. Developing tools to ease the switch, by providing appropriate information for the transition and indicating which units are now being controlled (possibly in a temporal highlighting fashion), should help prevent the loss of SA.

Another key point is that the player feels more "connected" when there is a one-to-one connection (the game is more engaging or fun). This suggests a potential danger for multi-UAV operators in that they may feel more in control by observing/controlling one asset in particular and switching among others only as needed, when in fact they are actually in less control of the others. Additionally, the desire to remain in FPS/ direct control mode may lead to attentional tunneling.

5.7 Turn-Based Planning – *Strategic Conquest*

Turn-based games are not nearly as popular amongst video gamers as they used to be. A large part of this is the instant and continuous immersion and involvement capability with increased speed and graphics processors in computers and consoles. This does however lead to a preponderance of emphasis on immediate interactions and split-second decision-making. While these are of course necessary skills for the control of military systems, there is also value in planning and having a proper tool to facilitate such planning. *Strategic Conquest* is an old game that stresses detailed planning over quick reflexes. In *Strategic Conquest*, the player must explore and conquer enemy and neutral cities in the game world using a variety of different assets with widely different capabilities (see Figure 15). Depending on the asset desired, they can take many game turns to produce and then use. Similarly, since movement takes a long time, desired units can easily be "out of position" when either planning or responding to an attack. The player's sense of immediate timing and manual dexterity do not play a part at all, since for each round, the player elects to attack or move away, and the result is decided randomly based on the statistics of the asset.

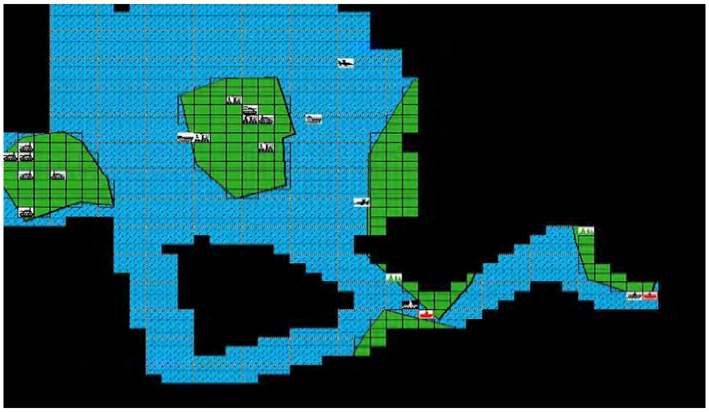

Figure 15. Strategic Conquest screen capture showing active battles in red, and unexplored regions in black[18].

Strategic Conquest is one of the only games in the list with planning and general risk assessment as the main hallmarks of game-play, with the "fog of uncertainty," or unexplored regions clearly entering into the process as well. This game may be the best representation of supervisory-level of control of multiple UAVs, and the means for which the game uses to alert the operator (e.g., flashing icons and auto-jump to attack a unit), are designed to improve the SA of the player. In addition to immediate concerns (i.e., attacks), the player needs to keep track of fuel consumption and the location of the many units. The techniques a player uses and the success of their own SA building strategies are a ripe area for multi-UAV research.

6.0 CONCLUSIONS

In this paper, we have described several kinds of video games and ways in which their design concepts may either directly inform or raise research questions for the area of unmanned system research, since essentially all unmanned system control will be performed through some sort of computer interface with semiautonomous (simulated intelligence) entities under the operator's control. The primary game types studied were arcade-type games, which have very specific interaction features, but are often abstract in both presentation and concept. These games provide some insight on basic operator principles such as performance failure. Another common game type is the first-person game, where the operator interacts through a virtual agent, with flight simulators and FPS being the most common. These provide a number of interface concepts, but not much in terms of autonomous system interaction. Finally, the richest area for games that could impact autonomous system design was found in RTS games, where the operator controls a large number of autonomous agents with varying capabilities. A number of operator alerting and multi-system control concepts can be found among the many games in this genre.

While the assessment of these various games led to various potential research areas, the similarities among many types of games, and the shortage of games that address some of the key problems with unmanned systems (mainly switching between the supervisory role and the direct control role), made the assessment a less direct utility than was originally desired. A probable reason for this relates to the differing primary goals of games and real unmanned system control. In games, the primary goal is player enjoyment, which includes a fairly continuous sense of player engagement and perceived impact on the game world, such that the interface is optimized

for continuity even at the expense of “mission performance” (e.g., *World of Warcraft*, above). This is contrasted with unmanned system control where the goal is successful mission accomplishment. Obtaining this goal is facilitated by reducing operator workload, such that the operator can control systems in an unstressed manner while maintaining enough control to preserve SA. Periods of inactivity are *not* penalized in control of real, unmanned systems, and delayed or even no direct response to actions is expected (e.g., images collected as a result of the mission are not always fed directly back into a subsequent mission). These fundamental differences make it somewhat difficult to directly translate successful concepts from one domain to the other. Admittedly, this may be due to the broad “survey” nature of the assessment where a large number of games were only superficially studied. A more targeted and more thorough analysis, possibly directed by the categories developed for the database, could be used to better address successful and unsuccessful concepts, particularly the “why” of its success or failure (much like the discussion of the *Raven Squad* issues). Allowing experienced gamers the opportunity to enter data concerning the games, either through direct interaction or through an interview process, will provide more relevant and more thorough data for performing subsequent statistical analyses of the concepts and driving future research areas.

7.0 REFERENCES

[1] Radio Airborne Mapping (RAM) Tracker, URL: http://www.ramsystemsllc.com/UAV.htm. Accessed May 4, 2011.

[2] OneSystem Universal Ground Control Station (UGCS), URL: http://www.defenseindustry.com/One-for-All-AAI-Textronics-UAV-Control-System-05412. Accessed May 4, 2011.

[3] Missile Command, URL: http://www.hollywood.com/news/Missile_Command_Coming_To_The_Big_Screen/7747897. Accessed May 4, 2011.

[4] Tetris, URL: http://www.thetetisgame.com. Accessed May 4, 2011.

[5] Starcraft, URL: http://www.gossipgamers.com/starcraft-course-opened-for-college-earn-credits/. Accessed May 4, 2011.

[6] Starcraft User Manual.

[7] Mech Commander, URL: http://games.softpedia.com/progScreenshots/Mech-Commander-Gold-No-Cd-Patch-Screenshot-53836.html. Accessed May 4, 2011.

[8] World of Warcraft, URL: http://log.archpaper.com/wordpress/archives/date/2009/04/page/3. Accessed May 4, 2011.

[9] Decursive Description, URL: http://www.2072productions.com/to/decursive.php. Accessed May 4, 2011.

[10] Decursive Overlay, URL: http://wow.joystiq.com/2008/04/18/addon-spotlight-decursive-2-0. Accessed May 4, 2011.

[11] Supreme Commander, URL: http://pc.ign.com/articles/929/929907p1.html. Accessed May 4, 2011.

[12] Supreme Commander, URL: http://artechnia.com/civis/viewtopic.php?f=22&t=236328&p=5683578. Accessed May 4, 2011.

[13] Supreme Commander, URL: http://www.cheatcodesgalore.com/pc/games/Supreme_Commander/Supreme_Commander-s2.jpg. Accessed May 4, 2011.

[14] Raven Squad Review, URL: http://www.gamespot.com/pc/action/ravensquadhiddendagger/review.html. Accessed May 4, 2011.

[15] Raven Squad Review, URL: http://xbox360.ign.com/articles/102/1021231p1.html. Accessed May 4, 2011.

[16] Raven Squad, URL: http://www.destructoid.com/review-raven-squad-147718.phtml. Accessed May 4, 2011.

[17] Raven Squad, URL: http://www.destructoid.com/review-raven-squad-147718.phtml. Accessed May 4, 2011.

[18] Strategic Conquest, URL: http://www.answers.com/topic/strategic-conquest. Accessed May 4, 2011.

APPENDIX – Description of the Categories in the Database

Basic information

System
This is the name of the game being analyzed. Different versions of the game may either be separate or lumped together depending on how different the versions are.

Year
This is the year in which the game was first released (or when the game was re-released if there were significant changes).

Website
The most relevant website where further information can be found about the game. Depending on the age of the game, this may or may not be the manufacturer's website.

Reason chosen
This is the reason the game was selected for inclusion in the database. This is provided mostly as a point of reference for why the game was included.

- Personal Experience
  - The author has previous experience with the game
- Suspected relevance
  - Either from the name or game type
- Recommendation
  - From a colleague or a review
- Best of ...
  - From a published "Best of…" list[1,2,3]

Highlight
This field indicates a game that is highlighted for further analysis in this document.

- Yes
- No

Physical system
The physical system indicates on what hardware the game is played. This may dictate various design and interface choices, so it is included as a separate category.

- Coin Op
  - Coin operated video game. Many are now available in newer or "retro editions"
- Console
  - E.g., X-Box, Playstation
- PC – Mac, PC
- Browser
  - Java-based, platform independent games
- iPhone
  - Often take advantage of motion/tipping

Game Type

More detailed descriptions of the game types with some different categorizations can be found here: http://en.wikipedia.org/wiki/Video_game_genres.

- First Person Shooter
  - A shooter game from the perspective of the shooter
  - E.g., Doom
- Third Person Shooter
  - A shooting or fighting game, usually from a three quarter or variable eyepoint view
  - E.g., Starfox
- Strategy - Turn-based
  - A game where a series of decisions are made, e.g., moving units, and until the player "submits" the choices, the game doesn't progress
  - E.g., Strategic Conquest
- Strategy - Real Time
  - A strategy game where decisions are made in the real time of the game. There are often two levels a development/management level (exploring and building new facilities) and combat, which occur simultaneously
  - E.g., Starcraft
- Board game
  - A computer game that emulates a board game
  - E.g., Archon
- 2D Side View
  - A game, often a jumping, maneuvering game where the view is fixed at a side view to the action, following the main character of the game. These are also referred to as "side-scrolling" games
  - E.g., Super Mario Bros
- 2D Top View
  - A game with a fixed top-down or "God's Eye" view of the scene. The player usually doesn't have independent control of the camera position
  - E.g., VASL – Virtual Advanced Squad Leader
- Driving - First Person
  - Games where the player is driving as if behind the wheel of a vehicle
  - E.g., F1 Challenge '99-'02
- Driving - Third Person
  - Games where the player drives the vehicle from a different perspective, usually directly behind the vehicle. Third person driving games usually can switch between first and third person
  - E.g., Pole Position
- Flight Sim - Realistic Commercial
  - Flight simulator of a commercial craft, such as small Cessnas up to Jumbo Jet. Attention is paid to proper take-off, flight and landing with as realistic a sense of control as possible
  - E.g., Flight Simulator X
- Flight Sim - Realistic Military
  - These flight sims often focus on combat, realistic missions and fast-paced action

    - E.g., Joint Strike Fighter
- Flight Sim – Futuristic
    - These often feature currently “impossible” flying techniques, such as complete VTOL control, and like military flight sims are often combat/mission directed
    - E.g., Descent II
- Management
    - In these games, there may or may be not an opponent with the possibility of combat, but rather the emphasis is on world building and decision making associated with it
    - E.g., Sim City
- RPG – MMORPG
    - A game in which the world persists and evolves with or without the player’s interaction on a continuous basis. Many players can be playing at the same time in the same world simultaneously
    - E.g., World of Warcraft
- RPG- Story-driven
    - Similar to third person shooters, and MMORPGs in their style of game combat, but the emphasis is on a definite scripted story for advancement
    - E.g., Baldur’s Gate II

Relevance

- Overall
    - The game in general is relevant, usually for a number of points
- Genre
    - Military and/or flight-based games fit this description
- Visuals
    - Either the video or the display itself are considered
- Dynamics
    - The dynamics of team or autonomous interactions is considered here

Interface Concepts

Input Device

The type of input device for interacting with the game.

- Joystick
    - Based on flight joysticks, but may be as simple as a three degree of freedom controller
- Paddle
    - Single degree of freedom rotating controller
- Keyboard
    - Includes a mouse
- Joypad
    - E.g., A playstation/X-box controller
- Accelerometer
    - E.g., A Wii Controller
- Touch

    - Touchscreens on iPhone games
- IR tracking
    - Infrared tracking device, e.g., http://www.naturalpoint.com/trackir/
- Voice
    - Speech controls and/or interactions through a headset

Display Concepts

- Lower Area
    - The information available to the player is fixed at the bottom portion of the screen
- Overlay
    - Information either indirectly or directly related to the main video window is overlaid on the video image. E.g., status information of units, overhead map
- Small Map
    - A small map which shows either a larger portion or the extent of a game world, often with the current view highlighted on the small map
- Variable
    - Any combination of the concepts above or others, but with the ability to toggle and/or modify the placement and presence of the display items

Notifications

These are the kinds of alerts that are given during the game.

- Speech
    - Ranging from simple status, e.g., “Ready” to more complex, e.g., “Sgt Smith is under attack”, alerts presented through speech
- Spatial Audio
    - With a stereo headset, indicators of the direction of the alert through sound
- Spatial Video
    - Visual indicators of the direction of the alert, e.g., arrows, specific highlighting
- Visual – persistent
    - A notification that appears, usually in a dedicated area
- "Flash" alerts
    - A notification that could be of any type, but it draws attention to itself through some kind of “flashing”, either visual or auditory
- Text Context alerts
    - The alert is given through a text message, either by simple words, e.g., “Disabled” or phrases “We must take the fort”

Teaming Options

The following indicates whether the game utilizes teaming, either with autonomous systems or with other players.

- None
- w/Autonomous
    - The players controls the autonomy at any range from a supervisory commander to the leader of a small squad
- w/Other players

- The players can team together either at exactly the same level of interaction or through a game structured hierarchy.

Autonomy Concepts
The following are concepts that relate to autonomous behavior in the games.

Autonomy

- None
  - No autonomous behavior
- Reactive Only
  - Units may only "shoot back" but otherwise have no autonomous behavior
- Simple "Waypoint" Tasking
  - Often a point and click "go here" type of interface
- Complex mission tasking
  - Higher level or stringing together of simpler commands, e.g., "defend the perimeter"
- Mid-mission decision making
  - The ability to change tasking mid-way, but still maintain most aspects of the original mission, as opposed to a complete reset of the mission.
- Discernable levels
  - Certain units are obviously more or less capable than others.
- Autonomy Settings
  - The player may adjust the autonomy levels for various units, to make the gameplay easier or harder
- Operator Control Switching
  - The ability to switch between direct control and autonomous/plan-based control
- Mixed Initiative Awareness
  - Indicator that the player or the autonomy is controlling a unit/entity

Modes
This section indicates the type of autonomous modes available.

- Planning
  - The player determines where the unit will go either through a simple interface or through more elaborate tasking. Usually for elaborate tasking, this is done in a turn-based fashion
- Direct Mobility Control
  - The player gives movement commands, but the autonomy actually controls the movement e.g., avoids obstacles, adjust speed
- Payload Control
  - The player can allow autonomous, often reactive, control of payload and still maintain direct control when needed
- Multiple Assets
  - The player is capable of controlling multiple and perhaps multiple types of units/entities simultaneously

Mode Awareness
This section indicates whether the game provides clues for whether the autonomy or the player is in charge of the unit/entity.

- Visual Indicator
    - Some sort of indicator, e.g., color change, grayed out icons
- Behavior
    - Entity will not respond appropriately or the player assuming control will override the autonomy directly
- "Switch" indicator
    - An indicator alerts the operator that the control has switched from one state to another, but there is no persistent indicator

Interaction Simultaneity
This section indicates how the operator interacts with the game's autonomy.

- Sequential control
    - The operator interacts with unit, usually planning missions, during which the autonomy is idle for all units
- Parallel control
    - As above, but the operator can interact with (plan) new units while others are performing their autonomous missions
- Simultaneous Control
    - Players can interact with and direct the autonomous entity while it is performing its mission

Autonomy SA
This area indicates how information from the autonomous system is shared with the player.

- Shared information
    - If the autonomous unit has the knowledge, the player does. This is often how removing "fog of war" effects are achieved. The player has their SA immediately updated without the need for the unit to return to a point
- Shared view
    - The player can immediately switch their view to see what the autonomous unit is seeing.

Teaming Concepts

Teaming Approach

- Loose - informally agreed upon
    - There is no particular game structure to enforce teaming decisions, including joining and leaving a team
- Medium - some rules
    - There are certain rules, such as shared knowledge and certain game-play effects, e.g. no friendly fire, but no restrictions within the team itself
- Tight - strict rules about teams
    - The setup of the teams may have strict limitations, e.g., size, role of participants, and there may a predefined game enforced hierarchy which cannot be superseded.

Information sharing

- None
  - No gameplay enabled sharing
- Realtime
  - Information updates are immediately shared as they become available
- Create Restore/SA
  - A mechanism is in place to push or pull information

Team SA

- Shared information
  - Teammates can share information, either passively (always shared) or actively, (pushing/pulling information)
- Shared view
  - Teammates can either look through a teammates video or seen a version of their screen

Research Concepts

OODA mapping

This section wasn't currently used, but the concept is to indicate games that stress one of the components in Col. Boyd's OODA loop description.

- Observe
- Orient
- Decide
- Act

AAR

These are checked if the game offers any kind of summary or after action review.

- Post-mission feedback
  - Relating to mission and operator performance
- Real-time "Coaching"
  - E.g., popup suggestions relating to gameplay instructions
- Playback
  - Video playback of previous scenes
- Different Views
  - Showing different perspectives of the same scene, either in replay or real time
- Summary of results
  - Statistics relating to the mission, e.g., number of kills, number or enemy

Training

These are checked if the game offers any kind of training.

- Limited Mission
  - A simple "test mission" prior to the actual game
- "Part task" training

    - Only certain aspects of the interface are exercised
- Help system during game
    - Often a popup list of commands, mission review
- Info deck on friendly/enemy units
    - Capability (not just status) information on a selected unit

Operator Issues
These are issues outside the design of the game, but the game somehow brings up these control issues.

- Catastrophic failure
    - Operator performance suddenly collapses
- Trust in automation
    - The level of the operators trust in automation impacts the game
- Level of involvement/SA
    - The operators sense of involvement and their ability to generate and maintain SA are key aspects to the game

Difficulty
What is the main driving source of difficulty and increasing difficulty in the game.

- Level based
    - The difficulty changes once a level is completed, usually by more/varied situations
- Success based
    - Degrees of success impact later difficulty
- Time pressure tempo
    - The time allowed to do tasks decreases
- Enemy skill
    - Either AI or opposing player skill

Application Aids
These are specific user aids built into the game.

- Augmented reality
    - Enhancements to the “live” video, such as damage indicators and enemy location indicators
- Information overlays
    - Related information, such as maps heads up displays
- Information access
    - Changing the accessibility of information based on its relevance to the current situation

Efficiency
These choices reflect certain design decisions as they relate to efficiency.

- Redundant inputs
    - Different means to input same information, e.g., keyboard command and icon mouse click
- "Natural" inputs

    - E.g., speech, pointing
- SA/Choices tradeoff
    - Whether the interface makes a specific balance or a conscious tradeoff between total information presentation and information appropriate to maximize SA

Research Issue Flag

If this field is highlighted, then a potential research issue has been identified from the game.

- Yes
- No

Appendix References

[1] "Best of" List, URL: http://top100.ign.com/2008/. Accessed May 4, 2011.

[2] "Best of" List, URL: http://www.old-wizard.com/top-100-video-games-of-all-time. Accessed May 4, 2011.

[3] "Best of" List, URL: http://uk.videogames.games.yahoo.com/specials/100games/. Accessed May 4, 2011.

LIST OF SYMBOLS, ABBREVIATIONS AND ACRONYMS

AFRL	Air Force Research Laboratory
FPS	First-Person Shooter
HUD	Heads-Up Display
MMORPG	Massively Multiplayer Online Role Playing Games
RAM	Radio Airborne Mapping
RTS	Real-Time Strategy
SA	Situation Awareness
UAS	Unmanned Air Vehicle System
UAV	Unmanned Air Vehicle